

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-289861

(43)Date of publication of application : 05.11.1993

(51)Int.Cl. G06F 9/06
G06F 11/28
G06F 12/06
G06F 13/24

(21)Application number : 04-115386 (71)Applicant : CANON INC
(22)Date of filing : 08.04.1992 (72)Inventor : ARIMA KAZUNORI

(54) MEMORY CONTENTS CHANGING DEVICE

(57)Abstract:

PURPOSE: To provide the memory contents changing device by which a control program stored in a ROM of an information apparatus can be changed at a low cost and also easily.

CONSTITUTION: A CPU 1 executes an access of a change storage part 3 simultaneously with an access of a control program stored in a program ROM 2 and checks whether the address contents which are being executed are the contents to be changed or not. In the case of the address to be changed a buffer 13 is set to a high impedance state in accordance with an output from the change storage part 3 and a data output of the program ROM 2 is stopped and also by making a memory 14 active a soft interrupting instruction is outputted to a data bus 10.

CLAIMS

[Claim(s)]

[Claim 1] A memory content change device comprising:

The 1st memory measure that memorizes a control program of information machines and equipment etc.

The 2nd memory measure that memorizes information on whether have the same address space as this 1st memory measure and said program should be changed into an address corresponding to a predetermined address of this 1st memory measure.

A reading means which reads simultaneously a memory content of said 1st memory

measure and said 2nd memory measure.

A control means which replaces with an output from said 1st memory measure and outputs predetermined data when information read from the 2nd memory measure by this reading means is contents which should change said program of said 1st memory measure.

[Claim 2] The memory content change device according to claim 1 wherein said predetermined data is an interrupt instruction which executes an alteration program.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] This invention relates to the memory content change device which changes the control program memorized by ROM in information machines and equipment.

[0002]

[Description of the Prior Art] Generally the control program of information machines and equipment is memorized by ROM.

The contents cannot be changed.

When changing this program the method exchanged for ROM which memorized the program which changed this ROM and the method of transmitting the memory content of ROM to RAM and changing said program on RAM are taken.

[0003]

[Problem(s) to be Solved by the Invention] However in the method of exchanging ROM the time and effort of opening information machines and equipment removing ROM from a substrate and substituting with new ROM arises.

[0004] In the method of transmitting the contents of the ROM to RAM and changing them on RAM the capacity of ROM and RAM which has an equivalent amount of the above capacity are needed and when the module of a program is large RAM or two or more RAM with so big capacity must be used. Apart from RAM used for this program changes since CPU needs RAM for preservation of data or an operation an increase in cost is caused.

[0005] On the other hand when the capacity of RAM is restricted for every work the program in ROM must be transmitted to RAM and must be changed and it becomes a cause which causes delay of processing.

[0006] An object of this invention is to provide the memory content change device which can change easily [are low cost and] the control program memorized by ROM of information machines and equipment in view of the above-mentioned conventional problem.

[0007]

[Means for Solving the Problem]claim 1 written this invention is characterized by it having been alike and comprising the following.

To achieve the above objectsthe 1st memory measure that memorizes a control program of information machines and equipmentetc.

The 2nd memory measure that memorizes information on whether have the same address space as this 1st memory measureand said program should be changed into an address corresponding to a predetermined address of this 1st memory measure.

A reading means which reads simultaneously a memory content of said 1st memory measure and said 2nd memory measure.

A control means which replaces with an output from said 1st memory measureand outputs predetermined data when information read from the 2nd memory measure by this reading means is contents which should change said program of said 1st memory measure.

[0008]The invention according to claim 2 is characterized by said predetermined data being an interrupt instruction which executes an alteration program in said invention according to claim 1.

[0009]

[Function]Executing the control program memorized by the 1st memory measurethe invention according to claim 1 reads the information simultaneously memorized by the 2nd memory measurewhen these information is contents which should change said control programstops execution of said control program and outputs predetermined data to instead of.

[0010]The invention according to claim 2 outputs the interrupt instruction which executes an alteration program as said predetermined data.

[0011]

[Example]Hereafterthe example of this invention is described in detail based on a drawing.

[0012]The [1st example] Drawing 1 is a block diagram showing the outline composition of the memory content change device concerning the 1st example of this invention.

[0013]CPU1 in which the memory content change device of this example manages control of the whole deviceROM(henceforth "program ROM") 2 which memorizes the control program which CPU1 executesIt comprises the decoder 5 which outputs a chip select signal to the change storage parts store 3 which memorizes the information on the change part of program ROM2the patch storage parts store 4 which memorizes the contents of change of this change partprogram ROM2 and the change storage parts store 3and the patch storage parts store 4.

[0014]Program ROM2 is 64 K bytes of ROM.

The address space of the change storage parts store 3 is the same as that of

program ROM2 and it is 64K-bit ROM which memorized "1" to the address corresponding to the address with which program ROM2 should have namely change a depth of 1 bit to each address and for which "0" was memorized to the other address.

[0015] It is connected with program ROM2 the change storage parts store 3 and the patch storage parts store 4 via 16 bits of low rank A0-A15 of the address bus 6 which comprises an 18-bit signal wire and as for 2 bits of higher ranks of the address bus 6 CPU1 is connected with the decoder 5 via A17 and A16. It is connected to program ROM2 the change storage parts store 3 and the patch storage parts store 4 via the signal wires 7 and 8 respectively this decoder 5 activates the signal wires 7 and 8 when [said] A17 and A16 are "00" and it activates the signal wire 9 at the time of "01". [top 2-bit]

[0016] On the other hand CPU1 is connected with the buffer 13 later mentioned via the data bus 10 which comprises a 16-bit signal wire the memory 14 and said patch storage parts store 4.

[0017] The output side of the change storage parts store 3 is connected to the chip select input edge of the memory 14 and the input side of the NOT gate 11. The output side of the NOT gate 11 is connected to one input edge of AND gate 12 and the output side of the decoder 5 is connected to the input edge of another side of AND gate 12 via the signal wire 7. The output side of AND gate 12 is connected with the buffer 13 and the input side of the buffer 13 is connected with the output side of program ROM2.

[0018] The buffer 13 saves temporarily the contents of program ROM2 addressed via the address bus 6 and outputs the contents to the data bus 10 according to the output of AND gate 12. That is when the output of AND gate 12 is "1" said saved contents are outputted to the data bus 10 the output of the buffer 13 is made into a high impedance state at the time of "0" and the output of these contents is suspended.

[0019] The memory 14 stores 1 byte of the instruction code decided beforehand for example soft interrupt instruction and outputs this instruction code to the data bus 10 according to the output of the change storage parts store 3. That is when the output of the change storage parts store 3 is "1" a soft interrupt instruction is outputted to the data bus 10 the output of the memory 14 is made into a high impedance state at the time of "0" and the output of this command is suspended.

[0020] Next operation of this memory content change device is explained in detail using drawing 2 and drawing 3.

[0021] Drawing 2 shows a translation table with the address of the patch storage parts store 4 which memorizes the address and contents of change of the changed part of program ROM2 and as for drawing 3 in two [upper] the address and contents of program ROM2 and lower two show the address and the contents of the patch storage parts store 4.

[0022]FirstCPU1 controls by reading the contents of program ROM2changing the value of the address bus 6 one by one. The change storage parts store 3 is also accessed at the same time program ROM2 is accessedand the contents of the change storage parts store 3 are outputted to the NOT gate 11 and the memory 14. When the address of accessed program ROM2 is not a change partsince "0" is outputted from the change storage parts store 3the contents of program ROM2 are outputted on the data bus 10 through the buffer 13. On the other handwhen the address of a change part is accessedthe change storage parts store 3 outputs "1" to the NOT gate 11 and the memory 14The buffer 13 will be in a high impedance statethe output from program ROM2 is forbiddenand the contents of the memory 14i.e.a soft interrupt instructionare instead outputted on the data bus 10. If this soft interrupt instruction is executedCPU1 will perform interrupt processing. This processing is also beforehand memorized to program ROM2. RAM which is not illustrated is providedwhen an interrupt occursthe address of the program which the interrupt generated is saved at the stack pointer of this RAMand the address into which interruption went from the contents of stack PONITA is known. That isthe table of the contents shown in drawing 2 is stored in the patch storage parts store 4and CPU1 searches the correction place address of the table of drawing 2 from the contents of the above-mentioned stack pointerand it runs a correcting program.

[0023]This is concretely explained based on drawing 3. When correcting this to "AND ALBL"they make the 104th street of the change storage parts store 3 memorize "1"although the commands stored in the 104th street of program ROM2 are "OR ALBL." If CPU1 accesses the 104th street of program ROM2a soft interrupt will occur. When this interrupt occursthe address loaded on the stack by interrupt processingi.e.104is readand control of a program is made to shift to the address shown in that table as compared with the table shown in drawing 2. In this casea program progresses to the 10000th street. As for the 10000th streetit is shown that the patch storage parts store 4 was chosen at top 2 bits (1)and the address of the beginning of the patch storage parts store 4 was chosen in 16 bits (0000) of low ranks. The command of "AND ALBL" and the command of "JMP 106" are written in the address of the beginning of the patch storage parts store 4. That isafter performing the 10000th streetthe program is stored in the patch storage parts store 4 in order to return control of a program to the address 106. Howeverby the time he follows to the 10000th street amendment which sinks below a stack and is returned to the state before generatingit is necessary to carry out. Herethe address 106 is an address with which the command next to "ORALBL" is stored.

[0024]Although 1 byte of soft interrupt instruction was memorized in the memory 14 in this examplewhat is necessary is just the command in which the same processing as soft interruptionsuch as a command in which it is not restricted to this but CPU1 goes into the special modeis possible.

[0025]Although ROM which has 1 bit of depth to each address constituted the change

storage parts store 3 from this example it may replace with this and RAM may be used. In this case in order to write data in RAM while the means which writes in data is required for RAM it is necessary to change the internal logic of the decoder 5. For example further in order to access the change storage parts store 4 also when [of an address] A17 and A16 are "10" [top 2-bit] When it constitutes so that chip select signal CS of the change storage parts store 3 may output via the signal wire 8 and writing data in the change storage parts store 3 A17 and A16 are set to "10" and 2 bits of higher ranks of an address access them. Or when CPU1 is CPU with memory space and two kinds of space of an I/O field it is also possible to constitute so that a change storage parts store may be arranged on an I/O field only when outputting the selection signal of a memory or I/O via the signal wire 15 which connects CPU1 and the decoder 5 and writing in the change storage parts store 3. In this case the decoder 5 needs to decode not only including top 2 bits of an address but including the level of the selection signal line 15 of a memory or I/O. It is necessary to connect via a buffer so that the contents of the data bus 10 can write in the change storage parts store 3.

[0026] When it is necessary to change to no addresses of program ROM capacity of a change storage parts store can be made small. For example although this example explained the case where there was space of 64 K bytes of program ROM When it is not necessary to give change to all those addresses for example and only the amount of 32 K bytes of low rank needs to give change the capacity of the change storage parts store 3 can be 4 K bytes of a half. However in order to prevent an image from appearing in top 32 K bytes in this case only when accessing 32 K bytes of that low rank by the decoder 5 it is necessary to activate the signal wire 8.

[0027] When the contents of a program to change continue on a large scale it does not change for every single address but may be made to skip a part for a lower order bit. For example only when [it is made not to access] it is got blocked and accessed by the block in every 4 bytes it may be made for an interrupt to generate the address of A0 and A1 among 16 bits of low ranks. Thereby capacity of a change storage parts store can be made small.

[0028] The [2nd example] Drawing 4 is a block diagram showing the outline composition of the memory content change device concerning the 2nd example of this invention.

[0029] A different point from the 1st example mentioned above in this example In the 1st example it is the point which constituted the change storage parts store and the patch storage parts store from the 2nd example by one memory to being a memory of the depth whose change storage parts store 3 is 1 bit and being a memory of the depth whose patch storage parts store 4 is 8 bits by storing the information on a change storage parts store as 8 bit data. When it is possible to use the usual EPROM without using a memory with a depth of 1 bit by this composition and the memory space of program ROM2 is 64 K bytes the memory space of a change storage

parts store can be managed with 8 K bytes.

[0030] Hereafter this example is described based on drawing 4.

[0031] The block diagram of drawing 1 mentioned above in drawing 4 and a corresponding element attach identical codes and are shown.

[0032] The memory 40 which this example comprised said CPU1 said program ROM2 and a change storage parts store and a patch storage parts store and comprised an EPROM. The decoder 41 of the address bus 6 which controls top 2 bits of chip select signals by the value of A17 and A16. The selector 42 which chooses and outputs 13 bit A0–A12 or A3–A15 among lower bit A0–A15 of the address bus 6 which CPU1 outputs. When the data of a change storage parts store is outputted from the memory 40 it comprises the data selector 43 which chooses and outputs 1 bit of the 8 bit data.

[0033] CPU1 is connected to program ROM2 via lower bit A0–A15 of the address bus 1.

[0034] 2 bits of higher ranks of the address bus 6 are connected with the decoder 41 via A17 and A16. It is connected with the selector 42 via 13 bit A0–A12 and A3–A15 and CPU1 is connected with the data selector 43 via low rank triplet A0–A2.

[0035] It is connected with the selector 42 via the 12-bit address bus 48 and the memory 40 is connected with the data selector 43 and the buffer 44 via the data bus 49. The decoder 41 is connected with chip select end CS of program ROM2 via the signal wire 7. It is connected with the address input end of the most significant bit of the memory 40, the selector end of the selector 42 and chip select input edge CS of the data selector 43 via the signal wire 45. It is connected with chip select input edge CS of the memory 40 via the signal wire 46 and is connected with chip select input edge CS of the buffer 44 via the signal wire 47.

[0036] Next the control action which the decoder 41 performs is explained.

[0037] The decoder 41 When [of the address bus 6] top 2 bits of values of A17 and A16 are "00" the signal wires 7, 45 and 46 are made into a high level (henceforth "H" level) the signal wire 47 is made into a low (henceforth the "L" level) when the value of A17 and A16 is "01" the signal wire 45 is used as the "L" level and the signal wires 46 and 47 are used as "H" level. When the signal wire 45 is "H" level the selector 42 specifies the address of the change storage parts store of the memory 40 with the address which outputted A3–A15 to the address bus 48 and combined 1 bit on the signal wire 45 with these 13 bits as the most significant bit. The memory 40 outputs the contents of an address specified as the data bus 49. With the signal wire 45 of "H" level the data selector 43 becomes active and this data selector 43 outputs 1 bit of 8 bit data of a change storage parts store to the NOT gate 11 based on the value of low rank triplet A0–A2 of the address bus 6.

[0038] On the other hand when the signal wire 45 is the "L" level the selector 42 outputs A0–A12 to the address bus 48 and specifies the address of the patch storage parts store of the memory 40 and the memory 40 outputs patch data to the data bus

49. Since the signal wire 47 is "H" level when the signal wire 45 is the "L" level the buffer 44 becomes active and the data of the data bus 49 is outputted to the data bus 10 as it is.

[0039] Next the control action of this example is explained.

[0040] CPU1 performs control according to the control program of program ROM2 changing the value of the address bus 6 one by one. When CPU1 does not specify the address of the change part of program ROM2 since [of the address bus 6] A17 and A16 are "00" [top 2-bit] The signal wires 45 and 46 serve as "H" level the selector 42 chooses address A3-A15 and the memory 40 outputs the data of the change storage parts store corresponding to this address to the data selector 43. At this time the data selector 43 outputs 1 bit of 8 bit data of this change storage parts store to the NOT gate 11 based on the value of low rank triplet A0-A2 of the address bus 6. For example when A0-A2 is "000" the least significant bit of said 8 bit data and in the case of "111" the most significant bit is outputted.

[0041] When the output value of the data selector 43 is "0" the output value of the NOT gate 11 is "1" and since the signal wire 7 is "H" level the output value of AND gate 12 is set to "1" and it becomes active [the buffer 13]. Since the chip select signal inputted into the memory 14 is "0" the memory 14 becomes inactive.

Therefore the contents of program ROM2 are outputted to the data bus 10 as it is.

[0042] On the other hand since the output value of the NOT gate 11 is set to "0" when the output value of the data selector 43 is "1" the buffer 13 will be in a high impedance state and suspends the output to the data bus 10. On the other hand the memory 14 becomes active and outputs the soft interrupt instruction memorized to the data bus 10.

[0043] CPU1 will shift control to soft interrupt processing if this soft interrupt instruction is received. Based on the address which the interrupt generated i.e. the address which should make a program change the address with which an alteration program is memorized from the data table memorized by the patch storage parts store is searched with soft interrupt processing and processing which shifts to this address is performed by it.

[0044] Concretely it explains based on the example of drawing 2 and drawing 3 mentioned above.

[0045] If CPU1 outputs the 104th address of program ROM2 CPU1 will shift to soft interrupt processing and it will output the address 10000 after table reference of drawing 2. At this time since [of the address bus 6] A17 and A16 are "01" the decoder 41 uses the signal wire 45 as the "L" level and uses the signal wires 46 and 47 as "H" level. [top 2-bit] With the output of this decoder 41 the selector 42 chooses address A0-A12 of the address bus 6 and outputs it to the address bus 48 and the address 00000 of the memory 40 is read with the address which combined 1 bit on the signal wire 45 with this address A0-A12 as the most significant bit. That is the start address of a patch storage parts store is chosen and the "AND ALBL"

command shown in drawing 3 is outputted to the data bus 49. Since it becomes active [the buffer 44] with the output of the decoder 41 this command is outputted to the data bus 10 and CPU1 executes this command. Next CPU1 executes "JMP 106" command and it outputs the address 106 to the address bus 6. That is as for 2 bits of higher ranks of the address bus 6 A17 and A16 are set to "00" and CPU1 shifts to the control action of usual program ROM2.

[0046] Also when a change storage parts store and a patch storage parts store are constituted from one memory as mentioned above the same control action as the 1st example can be performed and it becomes possible to miniaturize apparatus.

[0047] In the 2nd example although EPROM was used as the memory 40 it is realizable similarly not only by this ROM but RAM.

[0048] The [3rd example] Drawing 5 is a block diagram showing the outline composition of the memory content change device concerning the 3rd example of this invention.

[0049] The block diagram of drawing 1 mentioned above in drawing 5 and a corresponding element attach and show identical codes.

[0050] In this example a different point from the 1st example mentioned above is a point which outputs two or more byte command by this example to the data outputted from the memory 14 in the 1st example being a 1-byte command.

[0051] As shown in drawing 5 this example adds RSFF (RS flip flop) 50 AND gates 51 and 52 and OR gate 53 DFF (D flip-flop) 54-56 and the NOT gate 57 to the composition of said 1st example.

[0052] The output of the change storage parts store 3 is connected to S input edge of RSFF 50 and the output of the decoder 5 is connected to R input edge via the signal wire 58. The outgoing end of RSFF 50 is connected to the input edge of the NOT gate 11 and one input edge of AND gate 51. The outgoing end of AND gate 52 is connected to the input edge of another side of AND gate 51. The above-mentioned signal wire 58 connects the decoder 5 and the RST (reset) end of the memory 14 further.

[0053] In CPU1 it is the clock signal CLK and address from a clock source (not shown). Latch Enabling signal ALE is generated and D input edge and C input side of DFF 56 are supplied respectively. Each outgoing end of DFF 54-56 is connected to the input side of OR gate 53 and the input side of the NOT gate 57 is connected to C input edges each of DFF 56 and 55 and the output side is connected to the input edge of DFF 54 respectively. The output side of OR gate 53 is connected to one input edge of AND gate 52 and the signal wire 7 is connected to the input edge of another side of AND gate 52.

[0054] Drawing 6 is a timing chart which shows the address timing of CPU1 and the output wave of OR gate 53.

[0055] CPU1 advances processing synchronizing with the clock signal CLK as shown in drawing 6 and it outputs an address synchronizing with signal ALE. Namely CPU1 reads data between T3 and T four cycle at the time of a read cycle. If these signals CLK

and ALE are inputted into DFF56 from OR gate 53 the output wave shown in drawing 6 will be outputted.

[0056] Hereafter operation of this example is explained in detail.

[0057] RSFF50 is set when the output of the change storage parts store 3 is "1" it outputs "1" when the signal wire 58 is set to "H" level it is reset and it outputs "0."

[0058] The decoder 5 uses the signal wire 58 as "H" level only when [of the address bus 6] top 2 bits of values of A17 and A16 are "11" and at the time of "00" and "01" like the 1st example the signal wires 7 and 8 are used as "H" level and it uses the signal wire 9 as "H" level respectively.

[0059] CPU1 performs control according to the control program memorized by program ROM2 and the data from the change storage parts store 3 is outputted to S input edge of RSFF50 in the meantime. When the changed part of program ROM2 is not accessed since S input of RSFF50 and R input are "0" an initial state (in this case it is 0) is outputted the buffer 13 becomes active and the contents of program ROM2 are outputted to the data bus 10.

[0060] On the other hand when the changed part of program ROM2 is accessed S input is set to "1" R input is set to "0" and RSFF50 outputs 1. The buffer 13 will be in a high impedance state with the output of this RSFF50 the output of the contents of program ROM2 is suspended and the output of the memory 14 is validated.

[0061] The memory 14 is a memory which memorizes two or more instruction codes and it is constituted so that it may output 1 byte at a time one by one whenever the output of AND gate 51 rises from "0" to "1." Dummy data is put into the first 1 byte of the memory 14 and an expected command is put in from 2nd henceforth. This is for preventing the reading error of CPU1.

[0062] Therefore since the same waveform as the output wave of OR gate 53 AND gate 51 is indicated to be by drawing 6 whenever CPU1 accesses program ROM2 is outputted the memory 14 outputs the memorized command to the data bus 10 one by one.

[0063] The instruction code (for example JUMP command) which shifts to soft interrupt processing mentioned above and the processing which performs same processing is memorized by the memory 14. In the processing after shift processing of the address bus 6 which sets A17 and A16 to "11" is performed H level signal from the decoder 5 is impressed to R input edge of RSFF50 and RSFF50 outputs an output "0" and suspends the data output of the memory 14. [top 2-bit] Simultaneously H level signal from the decoder 5 is impressed to the RST end of the memory 14 via the signal wire 58 and a read-out order of the command memorized by this memory 14 is returned.

[0064] It is necessary to make the patch storage parts store 4 memorize the processing after said shift unlike each example mentioned above. For example when CPU1 receives a JUMP command the buffer 13 is still a high impedance state and it is because the release processing is performed by the processing after shifting so the

address in program ROM2 cannot be chosen. However if the command of the address bus 6 which sets A17 and A16 to "11" is added to the end of the command memorized by the memory 14 when CPU1 is CPU which predicts an instruction code it will become possible to store the processing after said shift in program ROM2. [top 2-bit]

[0065] Even if the command memorized by the memory 14 as mentioned above is two or more byte command it becomes possible to perform an effective control action.

[0066]

[Effect of the Invention] The 1st memory measure that memorizes the control program of information machines and equipment etc. in this invention as explained above The 2nd memory measure that memorizes the information on whether have the same address space as this 1st memory measure and said program should be changed into the address corresponding to the predetermined address of this 1st memory measure Said 1st memory measure and the reading means which reads the memory content of said 2nd memory measure simultaneously When the information read from the 2nd memory measure by this reading means is contents which should change said program of said 1st memory measure it has a control means which replaces with the output from said 1st memory measure and outputs predetermined data.

Therefore the effect it becomes possible to change easily [are low cost and] the control program memorized by ROM of information machines and equipment is done so.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is a block diagram showing the outline composition of the memory content change device concerning the 1st example of this invention.

[Drawing 2] It is an explanatory view showing the table of the data memorized by program ROM of the 1st example.

[Drawing 3] It is an explanatory view showing the data and correct data of program ROM of the 1st example.

[Drawing 4] It is a block diagram showing the outline composition of the memory content change device concerning the 2nd example of this invention.

[Drawing 5] It is a block diagram showing the outline composition of the memory content change device concerning the 3rd example of this invention.

[Drawing 6] It is a timing chart which shows the address timing of CPU and the output wave of an OR gate.

[Description of Notations]

1 CPU (reading means)

2 Program ROM (the 1st memory measure)

3 Change storage parts store (the 2nd memory measure)

14 Memory (control means)

(19)日本国特許庁(J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平5-289861

(43)公開日 平成5年(1993)11月5日

| (51)Int.Cl. ⁵ | 識別記号 | 庁内整理番号 | F I | 技術表示箇所 |
|--------------------------|---------|---------|-----|--------|
| G 0 6 F 9/06 | 4 4 0 N | 8944-5B | | |
| 11/28 | 3 3 0 B | 9290-5B | | |
| 12/06 | 5 2 0 | 8841-5B | | |
| 13/24 | 3 1 0 C | 9072-5B | | |

審査請求 未請求 請求項の数2(全 9 頁)

(21)出願番号 特願平4-115386

(22)出願日 平成4年(1992)4月8日

(71)出願人 000001007

キャノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 有馬 和範

東京都大田区下丸子3丁目30番2号 キャ
ノン株式会社内

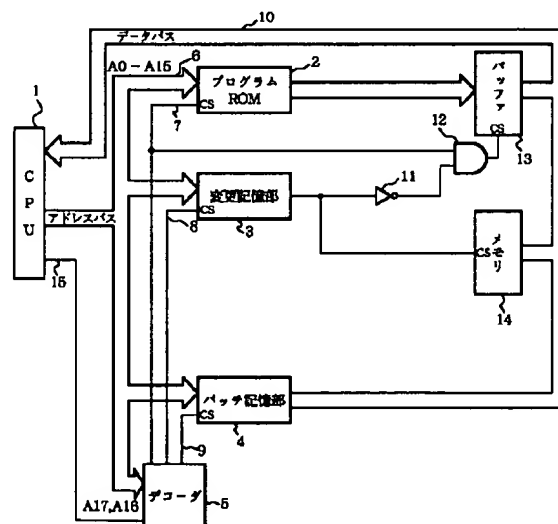
(74)代理人 弁理士 渡部 敏彦

(54)【発明の名称】 メモリ内容変更装置

(57)【要約】

【目的】 情報機器のROMに記憶された制御プログラムを低コストで且つ容易に変更することが可能なメモリ内容変更装置を提供する。

【構成】 CPU1は、プログラムROM2に記憶された制御プログラムのアクセスと同時に変更記憶部3のアクセスを行い、実行中のアドレス内容が変更すべき内容か否かをチェックする。変更すべきアドレスである場合、変更記憶部3からの出力に応じてバッファ13をハイインピーダンス状態にしてプログラムROM2のデータ出力を停止するとともに、メモリ14をアクティブにしてデータバス10にソフト割込み命令を出力する。



【特許請求の範囲】

【請求項 1】 情報機器の制御プログラム等を記憶する第 1 の記憶手段と、該第 1 の記憶手段と同一のアドレス空間を有し、該第 1 の記憶手段の所定のアドレスに対応するアドレスに前記プログラムを変更すべきか否かの情報を記憶する第 2 の記憶手段と、前記第 1 の記憶手段と前記第 2 の記憶手段の記憶内容を同時に読み出す読み出し手段と、該読み出し手段により第 2 の記憶手段から読み出された情報が前記第 1 の記憶手段の前記プログラムを変更すべき内容である場合、前記第 1 の記憶手段からの出力に代えて所定のデータを出力する制御手段とを有することを特徴とするメモリ内容変更装置。

【請求項 2】 前記所定のデータは変更プログラムを実行する割り込み命令であることを特徴とする請求項 1 記載のメモリ内容変更装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、情報機器において ROM に記憶された制御プログラムを変更するメモリ内容変更装置に関する。

【0002】

【従来の技術】一般に、情報機器の制御プログラムは、ROM に記憶されており、その内容を変更することは出来ない。かかるプログラムを変更する場合、該 ROM を変更したプログラムを記憶した ROM と交換する方法や、ROM の記憶内容を RAM に転送し、RAM 上で前記プログラムを変更する方法が採られている。

【0003】

【発明が解決しようとする課題】しかしながら、ROM を交換する方法では、情報機器を開封して ROM を基板から取り外して新しい ROM と差し替えるという手間が生じる。

【0004】また、ROM の内容を RAM に転送して RAM 上で変更する方法では、ROM の容量と同量以上の容量を有する RAM を必要とし、プログラムのモジュールが大きい場合にはそれだけ容量の大きな RAM または複数個の RAM を使用しなければならない。更に、このプログラム変更に用いる RAM とは別に、CPU はデータの保存あるいは演算等のために RAM を必要とするためコストの増加を招く。

【0005】一方、RAM の容量に限られる場合には、作業毎に ROM 中のプログラムを RAM に転送して変更しなければならず、処理の遅延を来す原因となる。

【0006】本発明は、上記従来の問題点に鑑み、情報機器の ROM に記憶された制御プログラムを低コストで且つ容易に変更することが可能なメモリ内容変更装置を提供することを目的とする。

【0007】

【課題を解決するための手段】請求項 1 記載の発明は、上記目的を達成するために、情報機器の制御プログラム

等を記憶する第 1 の記憶手段と、該第 1 の記憶手段と同一のアドレス空間を有し、該第 1 の記憶手段の所定のアドレスに対応するアドレスに前記プログラムを変更すべきか否かの情報を記憶する第 2 の記憶手段と、前記第 1 の記憶手段と前記第 2 の記憶手段の記憶内容を同時に読み出す読み出し手段と、該読み出し手段により第 2 の記憶手段から読み出された情報が前記第 1 の記憶手段の前記プログラムを変更すべき内容である場合、前記第 1 の記憶手段からの出力に代えて所定のデータを出力する制御手段とを有することを特徴とする。

【0008】請求項 2 記載の発明は、前記請求項 1 記載の発明において、前記所定のデータは変更プログラムを実行する割り込み命令であることを特徴とする。

【0009】

【作用】請求項 1 記載の発明は、第 1 の記憶手段に記憶された制御プログラムを実行しながら、同時に第 2 の記憶手段に記憶された情報を読み出し、該情報が前記制御プログラムを変更すべき内容である場合、前記制御プログラムの実行を中止し、代わりに所定のデータを出力する。

【0010】請求項 2 記載の発明は、前記所定のデータとして変更プログラムを実行する割り込み命令を出力する。

【0011】

【実施例】以下、本発明の実施例を図面に基づき詳細に説明する。

【0012】〔第 1 実施例〕図 1 は、本発明の第 1 実施例に係るメモリ内容変更装置の概略構成を示すブロック図である。

【0013】本実施例のメモリ内容変更装置は、装置全体の制御を司る CPU 1 と、CPU 1 が実行する制御プログラムを記憶する ROM（以下「プログラム ROM」と云う）2 と、プログラム ROM 2 の変更個所の情報を記憶する変更記憶部 3 と、該変更個所の変更内容を記憶するパッチ記憶部 4 と、プログラム ROM 2、変更記憶部 3 及びパッチ記憶部 4 へチップセレクト信号を出力するデコーダ 5 とから構成される。

【0014】プログラム ROM 2 は、64 K バイトの ROM であり、変更記憶部 3 は、アドレス空間がプログラム ROM 2 と同一で、各アドレスに対して 1 ビットの深さを有する、即ち、プログラム ROM 2 の変更すべきアドレスに対応するアドレスに“1”、それ以外のアドレスに“0”を記憶した 64 K ビットの ROM である。

【0015】CPU 1 は、18 ビットの信号線から成るアドレスバス 6 の下位 16 ビット A0-A15 を介してプログラム ROM 2、変更記憶部 3 及びパッチ記憶部 4 と接続され、アドレスバス 6 の上位 2 ビット A17、A16 を介してデコーダ 5 と接続されている。このデコーダ 5 は、信号線 7、8、9 を介してプログラム ROM 2、変更記憶部 3 及びパッチ記憶部 4 に夫々接続され、

前記上位2ビットA17, A16が“00”の時信号線7, 8をアクティブにし、“01”の時信号線9をアクティブにする。

【0016】一方、CPU1は、16ビットの信号線から成るデータバス10を介して後述するバッファ13、メモリ14及び前記パッチ記憶部4と接続されている。

【0017】また、変更記憶部3の出力側は、メモリ14のチップセレクト入力端及びNOTゲート11の入力側に接続され、NOTゲート11の出力側はANDゲート12の一方の入力端に接続され、ANDゲート12の他方の入力端には信号線7を介してデコーダ5の出力側が接続されている。更に、ANDゲート12の出力側はバッファ13と接続され、バッファ13の入力側はプログラムROM2の出力側と接続されている。

【0018】バッファ13は、アドレスバス6を介してアドレス指定されたプログラムROM2の内容を一時的に保存し、ANDゲート12の出力に応じてその内容をデータバス10に出力する。即ち、ANDゲート12の出力が“1”の時前記保存された内容をデータバス10に出力し、“0”の時バッファ13の出力をハイインピーダンス状態にして該内容の出力を停止する。

【0019】メモリ14は、あらかじめ決められた命令コード、例えば1バイトのソフト割込み命令を格納し、変更記憶部3の出力に応じて該命令コードをデータバス10に出力する。即ち、変更記憶部3の出力が“1”の時ソフト割込み命令をデータバス10に出力し、“0”の時メモリ14の出力をハイインピーダンス状態にして該命令の出力を停止する。

【0020】次に、図2及び図3を用いて本メモリ内容変更装置の動作を詳細に説明する。

【0021】図2は、プログラムROM2の変更部分のアドレスとその変更内容を記憶するパッチ記憶部4のアドレスとの変換テーブルを示し、図3は上の2つがプログラムROM2のアドレスとその内容、下の2つがパッチ記憶部4のアドレスとその内容を示す。

【0022】まず、CPU1はアドレスバス6の値を順次変更しながら、プログラムROM2の内容を読み込み、制御を行う。プログラムROM2がアクセスされると同時に変更記憶部3もアクセスされ、変更記憶部3の内容がNOTゲート11及びメモリ14に出力される。アクセスされたプログラムROM2のアドレスが変更箇所でないときは、変更記憶部3から“0”が出力されるのでバッファ13を通じて、プログラムROM2の内容がデータバス10上に出力される。一方、変更箇所アドレスがアクセスされたとき、変更記憶部3は“1”をNOTゲート11及びメモリ14に出力し、バッファ13はハイインピーダンス状態となりプログラムROM2からの出力を禁止し、代わりにメモリ14の内容、即ちソフト割込み命令がデータバス10上に出力される。該ソフト割込み命令が実行されると、CPU1は割込み処

理を行なう。この処理もあらかじめプログラムROM2に記憶してある。図示しないRAMが設けられ、割込みが発生すると割込みが発生したプログラムのアドレスが該RAMのスタックポイントに保存され、スタックポイントの内容から割込みが入ったアドレスが分かるようになっている。即ち、図2に示される内容のテーブルがパッチ記憶部4に格納され、CPU1は前述のスタックポイントの内容から図2のテーブルの修正先アドレスを検索し、修正プログラムを走らせる。

【0023】これを図3に基づき具体的に説明する。プログラムROM2の104番地に格納されている命令は、“OR AL, BL”であるが、これを“AND AL, BL”に修正する場合、変更記憶部3の104番地に“1”を記憶させておく。CPU1がプログラムROM2の104番地をアクセスするとソフト割込みが発生する。この割込みが発生すると、割込み処理でスタックにつまれたアドレス、即ち104を読み込み、図2に示されるテーブルと比較し、そのテーブルに示されるアドレスにプログラムの制御を移行させる。この場合、プログラムは10000番地に進む。10000番地は、上位2ビット(1)でパッチ記憶部4が選択され、下位16ビット(0000)でパッチ記憶部4の最初のアドレスが選択されたことが示される。パッチ記憶部4の最初のアドレスには、“AND AL, BL”の命令と、“JMP 106”の命令が書き込まれている。つまり、10000番地を実行した後、プログラムの制御をアドレス106に戻すべくパッチ記憶部4にプログラムが格納されている。ただし、スタックを割込み発生前の状態に戻す補正を10000番地に進むまでに行なっておく必要がある。ここで、アドレス106は、“OR AL, BL”の次の命令が格納されているアドレスである。

【0024】本実施例ではメモリ14に1バイトのソフト割込み命令を記憶したが、これに限られずCPU1が特殊なモードに入る命令等ソフト割込みと同様な処理が可能な命令であればよい。

【0025】また、本実施例では、変更記憶部3を各アドレスに対して1ビットの深みのあるROMで構成したが、これに代えてRAMを用いてもよい。この場合、RAMにデータを書き込むためには、RAMにデータを書き込む手段が必要であるとともに、デコーダ5の内部ロジックを変更する必要がある。例えば、変更記憶部4をアクセスするために更にアドレスの上位2ビットA17, A16が“10”の場合にも、信号線8を介して変更記憶部3のチップセレクト信号CSが出力するように構成し、変更記憶部3にデータを書き込む際には、アドレスの上位2ビットA17, A16を“10”にしてアクセスする。または、CPU1がメモリ空間とI/O空間の2種類の空間を持つCPUである場合は、CPU1とデコーダ5とを接続する信号線15を介してメモリか

1/Oかの選択信号を出力し、変更記憶部3に書き込みするときだけ1/O空間上に変更記憶部が配置されるように構成することも可能である。この場合は、デコーダ5がアドレスの上位2ビットだけでなくメモリか1/Oかの選択信号線15のレベルをも含めてデコードする必要がある。更に、データバス10の内容が変更記憶部3に書き込み出来るようにバッファを介して接続する必要がある。

【0026】また、プログラムROMのすべてのアドレスに対して変更する必要がある場合、変更記憶部の容量を小さくすることが出来る。例えば、本実施例では、64KバイトのプログラムROMの空間がある場合について説明したが、そのすべてのアドレスに変更を与える必要がなく、例えば、下位32Kバイト分しか変更を与える必要がない場合には変更記憶部3の容量を半分の4Kバイトにすることが出来る。しかし、この場合、上位32Kバイトにイメージが出るのを防止するため、デコーダ5でその下位32Kバイトをアクセスするときだけ信号線8をアクティブにする必要がある。

【0027】また、変更するプログラム内容が大規模に亘る場合には、1アドレス毎に変更するのではなく、下位数ビット分を端折るようにしてもよい。例えば、下位16ビットのうち、A0、A1のアドレスはアクセスしないようにする、つまり、4バイトごとのブロックにアクセスされたときのみ、割込みが発生するようにしてもよい。これにより、変更記憶部の容量を小さくすることが出来る。

【0028】〔第2実施例〕図4は、本発明の第2実施例に係るメモリ内容変更装置の概略構成を示すブロック図である。

【0029】本実施例において、上述した第1実施例と異なる点は、第1実施例では変更記憶部3が1ビットの深さのメモリでありパッチ記憶部4が8ビットの深さのメモリであるのに対して、第2実施例では変更記憶部の情報を8ビットデータとして格納することにより、変更記憶部とパッチ記憶部とを1つのメモリで構成した点である。この構成により、1ビットの深さを持つメモリを使用せずに通常のEPROMを使用することが可能であり、また、プログラムROM2のメモリ容量が64Kバイトである場合には、変更記憶部のメモリ容量は8Kバイトで済むこととなる。

【0030】以下、図4に基づいて本実施例を説明する。

【0031】尚、図4において上述した図1のブロック図と対応する要素は、同一符号を付して示してある。

【0032】本実施例は、前記CPU1と、前記プログラムROM2と、変更記憶部およびパッチ記憶部から成りEPROMで構成されたメモリ40と、アドレスバス6の上位2ビットA17、A16の値によりチップセレクト信号を制御するデコーダ41と、CPU1が出力す

るアドレスバス6の下位ビットA0-A15のうち13ビットA0-A12またはA3-A15を選択して出力するセクタ42と、メモリ40から変更記憶部のデータが出力された場合に、その8ビットデータのうち1ビットを選択して出力するデータセクタ43とから構成される。

【0033】CPU1は、アドレスバス1の下位ビットA0-A15を介してプログラムROM2に接続されている。

【0034】更に、CPU1は、アドレスバス6の上位2ビットA17、A16を介してデコーダ41と接続され、13ビットA0-A12及びA3-A15を介してセクタ42と接続され、下位3ビットA0-A2を介してデータセクタ43と接続されている。

【0035】また、メモリ40は、12ビットのアドレスバス48を介してセクタ42と接続され、データバス49を介してデータセクタ43及びバッファ44と接続されている。更に、デコーダ41は信号線7を介してプログラムROM2のチップセレクト端CSと接続され、信号線45を介してメモリ40の最上位ビットのアドレス入力端、セクタ42のセクタ端及びデータセクタ43のチップセレクト入力端CSと接続され、信号線46を介してメモリ40のチップセレクト入力端CSと接続され、信号線47を介してバッファ44のチップセレクト入力端CSと接続されている。

【0036】次に、デコーダ41が実行する制御動作について説明する。

【0037】デコーダ41は、アドレスバス6の上位2ビットA17、A16の値が“00”の場合、信号線7、45、46を高レベル（以下、“H”レベルという）、信号線47を低レベル（以下、“L”レベルという）にし、A17、A16の値が“01”の場合、信号線45を“L”レベル、信号線46、47を“H”レベルにする。信号線45が“H”レベルである時、セクタ42はアドレスバス48にA3-A15を出力してこの13ビットに信号線45上の1ビットを最上位ビットとして組合せたアドレスによりメモリ40の変更記憶部のアドレスを指定し、メモリ40はデータバス49に指定されたアドレス内容出力する。更に、“H”レベルの信号線45によってデータセクタ43がアクティブとなり、該データセクタ43はアドレスバス6の下位3ビットA0-A2の値に基づいて、変更記憶部の8ビットデータのうち1ビットをNOTゲート11に出力する。

【0038】一方、信号線45が“L”レベルである時には、セクタ42はアドレスバス48へA0-A12を出力してメモリ40のパッチ記憶部のアドレスを指定し、メモリ40はデータバス49にパッチデータを出力する。信号線45が“L”レベルのとき信号線47は“H”レベルであるため、バッファ44がアクティブと

なり、データバス49のデータはそのままデータバス10に出力される。

【0039】次に、本実施例の制御動作について説明する。

【0040】CPU1は、アドレスバス6の値を順次変更しながらプログラムROM2の制御プログラムに従って制御を実行する。CPU1がプログラムROM2の変更個所のアドレスを指定しない場合、アドレスバス6の上位2ビットA17、A16は“00”であるため、信号線45、46は“H”レベルとなってセレクタ42はアドレスA3-A15を選択し、メモリ40は該アドレスに対応した変更記憶部のデータをデータセレクタ43に出力する。このとき、データセレクタ43は、アドレスバス6の下位3ビットA0-A2の値に基づいて該変更記憶部の8ビットデータのうちの1ビットをNOTゲート11に出力する。例えば、A0-A2が“000”の場合には前記8ビットデータのうちの最下位ビット、“111”の場合には最上位ビットを出力する。

【0041】データセレクタ43の出力値が“0”である場合、NOTゲート11の出力値は“1”であり、また、信号線7は“H”レベルであることからANDゲート12の出力値が“1”となりバッファ13はアクティブとなる。また、メモリ14に入力されるチップセレクト信号が“0”であるためメモリ14は非アクティブとなる。したがって、データバス10にはプログラムROM2の内容がそのまま出力される。

【0042】一方、データセレクタ43の出力値が“1”である場合、NOTゲート11の出力値が“0”になるため、バッファ13はハイインピーダンス状態になってデータバス10に対する出力を停止する。一方、メモリ14はアクティブとなり、記憶されているソフト割込み命令をデータバス10に出力する。

【0043】CPU1は、このソフト割込み命令を受信するとソフト割込み処理に制御を移行する。ソフト割込み処理では、割込みが発生したアドレス、即ちプログラム変更をすべきアドレスに基づいて、パッチ記憶部に記憶されるデータテーブルから変更プログラムの記憶されるアドレスを検索し、該アドレスに移行する処理を実行する。

【0044】具体的に、前述した図2、図3の例に基づいて説明する。

【0045】CPU1がプログラムROM2の104番地のアドレスを出力すると、CPU1はソフト割込み処理に移行し、図2のテーブル参照後アドレス10000を出力する。このとき、アドレスバス6の上位2ビットA17、A16は“01”であるため、デコーダ41は信号線45を“L”レベル、信号線46、47を“H”レベルにする。このデコーダ41の出力により、セレクタ42はアドレスバス6のアドレスA0-A12を選択してアドレスバス48に出力し、該アドレスA0-A1

2に信号線45上の1ビットを最上位ビットとして組合せたアドレスによりメモリ40のアドレス00000を読み出す。即ち、パッチ記憶部の先頭アドレスが選択され、図3に示される“AND AL, BL”命令をデータバス49に出力する。デコーダ41の出力によりバッファ44はアクティブとなるため、該命令をデータバス10に出力し、CPU1は該命令を実行する。次に、CPU1は“JMP 106”命令を実行し、アドレスバス6にアドレス106を出力する。即ち、アドレスバス6の上位2ビットA17、A16が“00”となり、CPU1は通常のプログラムROM2の制御動作に移行する。

【0046】以上のようにして、変更記憶部とパッチ記憶部とを一つのメモリで構成した場合にも第1実施例と同様の制御動作を行うことができ、機器を小型化することが可能となる。

【0047】また、第2実施例では、メモリ40としてEPROMを用いたが、かかるROMに限らずRAMでも同様に実現できる。

【0048】[第3実施例] 図5は、本発明の第3実施例に係るメモリ内容変更装置の概略構成を示すブロック図である。

【0049】尚、図5において上述した図1のブロック図と対応する要素は同一符号を付して示す。

【0050】本実施例において、上述した第1実施例と異なる点は、第1実施例ではメモリ14から出力されるデータが1バイト命令であるのに対して、本実施例では複数バイト命令を出力する点である。

【0051】図5に示されるように、本実施例は、前記第1実施例の構成にRSFF(RSフリップフロップ)50と、ANDゲート51、52とORゲート53と、DFF(Dフリップフロップ)54~56と、NOTゲート57とを付加したものである。

【0052】RSFF50のS入力端には、変更記憶部3の出力が接続され、R入力端にはデコーダ5の出力が信号線58を介して接続されている。更に、RSFF50の出力端はNOTゲート11の入力端とANDゲート51の一入力端に接続されている。また、ANDゲート51の他方の入力端には、ANDゲート52の出力端が接続されている。上記信号線58は更にデコーダ5とメモリ14のRST(リセット)端とを接続する。

【0053】CPU1には、クロック源(図示しない)からのクロック信号CLKおよびアドレスラッチ許可信号ALEを発生し、それぞれDFF56のD入力端とC入力側に供給する。DFF54~56の各出力端はORゲート53の入力側に接続されると共に、NOTゲート57の入力側はDFF56、55の各C入力端に、出力側はDFF54の入力端にそれぞれ接続されている。ORゲート53の出力側はANDゲート52の一入力端に接続され、ANDゲート52の他方の入力端には

信号線7が接続されている。

【0054】図6は、CPU1のアドレスタイミングとORゲート53の出力波形とを示すタイミングチャートである。

【0055】CPU1は、図6に示されるようにクロック信号CLKに同期して処理を進め、信号ALEに同期してアドレスを出力する。即ち、CPU1はリードサイクルの時、T3とT4サイクルの間でデータを読み込む。この信号CLKとALEをDFF56に入力すると、ORゲート53からは図6に示す出力波形が出力される。

【0056】以下、本実施例の動作を詳細に説明する。

【0057】RSFF50は、変更記憶部3の出力が“1”のときセットされて“1”を出力し、信号線58が“H”レベルになったときリセットされて“0”を出力する。

【0058】デコーダ5は、アドレスバス6の上位2ビットA17、A16の値が“11”のときのみ信号線58を“H”レベルにし、“00”、“01”のときは、第1実施例と同様にそれぞれ信号線7、8を“H”レベル、信号線9を“H”レベルとする。

【0059】CPU1は、プログラムROM2に記憶された制御プログラムに従って制御を実行し、この間変更記憶部3からのデータはRSFF50のS入力端に出力される。プログラムROM2の変更部分がアクセスされない場合には、RSFF50のS入力、R入力共に“0”であるため初期状態（この場合0である）が出力され、バッファ13がアクティブとなってデータバス10にプログラムROM2の内容が出力される。

【0060】一方、プログラムROM2の変更部分がアクセスされた場合は、RSFF50はS入力が“1”、R入力が“0”となって1を出力する。このRSFF50の出力によりバッファ13はハイインピーダンス状態になり、プログラムROM2の内容の出力を停止し、メモリ14の出力を有効にする。

【0061】メモリ14は、複数の命令コードを記憶するメモリで、ANDゲート51の出力が“0”から“1”に立ち上がる度に1バイトずつ順次出力するように構成されている。メモリ14の最初の1バイトにはダミーデータを入れ、2番目以降から所期の命令を入れる。これは、CPU1の読み誤りを防止するためである。

【0062】したがって、CPU1がプログラムROM2をアクセスする度に、ANDゲート51が図6で示されるORゲート53の出力波形と同一の波形を出力するため、メモリ14は記憶された命令を順次データバス10に出力する。

【0063】メモリ14には、前述したソフト割込み処理と同様の処理を行う処理へ移行する命令コード（例えば、JUMP命令）が記憶される。移行後の処理では、

アドレスバス6の上位2ビットA17、A16を“11”にする処理を実行し、デコーダ5からの“H”レベル信号がRSFF50のR入力端に印加され、RSFF50が出力“0”を出力してメモリ14のデータ出力を停止する。また、同時にデコーダ5からの“H”レベル信号は、信号線58を介してメモリ14のRST端に印加されて、該メモリ14に記憶される命令の読み出し順序を元に戻す。

【0064】また、前記移行後の処理は、前述した各実施例と異なりパッチ記憶部4に記憶させる必要がある。例えば、CPU1がJUMP命令を受信した場合、バッファ13は未だハイインピーダンス状態であり、その解除処理を移行後の処理で行うため、プログラムROM2内のアドレスを選択できないからである。しかし、CPU1が命令コードを先読みするCPUである場合には、メモリ14に記憶された命令の最後にアドレスバス6の上位2ビットA17、A16を“11”にする命令を付加しておけば、前記移行後の処理をプログラムROM2に格納することが可能となる。

【0065】以上のようにして、メモリ14に記憶された命令が複数バイト命令であっても、有効な制御動作を実行することが可能となる。

【0066】

【発明の効果】以上説明したように、本発明は、情報機器の制御プログラム等を記憶する第1の記憶手段と、該第1の記憶手段と同一のアドレス空間を有し、該第1の記憶手段の所定のアドレスに対応するアドレスに前記プログラムを変更すべきか否かの情報を記憶する第2の記憶手段と、前記第1の記憶手段と前記第2の記憶手段の記憶内容を同時に読み出す読み出し手段と、該読み出し手段により第2の記憶手段から読み出された情報が前記第1の記憶手段の前記プログラムを変更すべき内容である場合、前記第1の記憶手段からの出力に代えて所定のデータを出力する制御手段とを有するので、情報機器のROMに記憶された制御プログラムを低コストで且つ容易に変更することが可能となる効果を奏する。

【図面の簡単な説明】

【図1】本発明の第1実施例に係るメモリ内容変更装置の概略構成を示すブロック図である。

【図2】第1実施例のプログラムROMに記憶されるデータのテーブルを示す説明図である。

【図3】第1実施例のプログラムROMのデータとその修正データを示す説明図である。

【図4】本発明の第2実施例に係るメモリ内容変更装置の概略構成を示すブロック図である。

【図5】本発明の第3実施例に係るメモリ内容変更装置の概略構成を示すブロック図である。

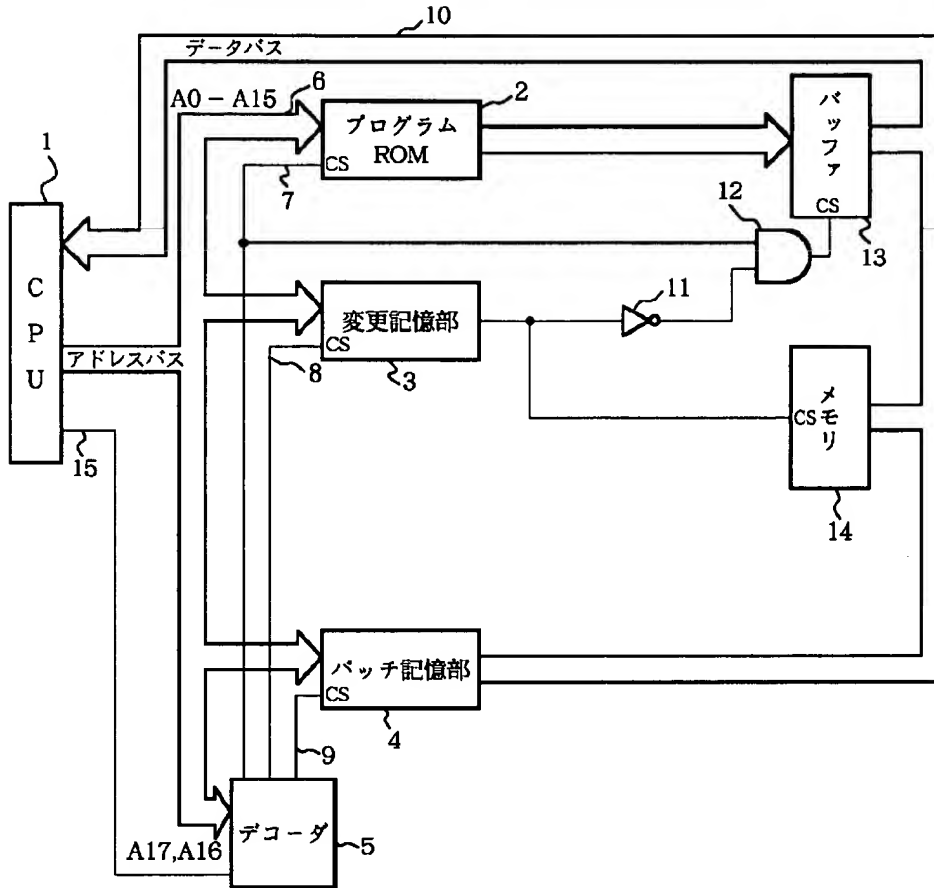
【図6】CPUのアドレスタイミングとORゲートの出力波形とを示すタイミングチャートである。

【符号の説明】

- 1 CPU (読み出し手段)
2 プログラムROM (第1の記憶手段)

- 3 変更記憶部 (第2の記憶手段)
14 メモリ (制御手段)

【図1】



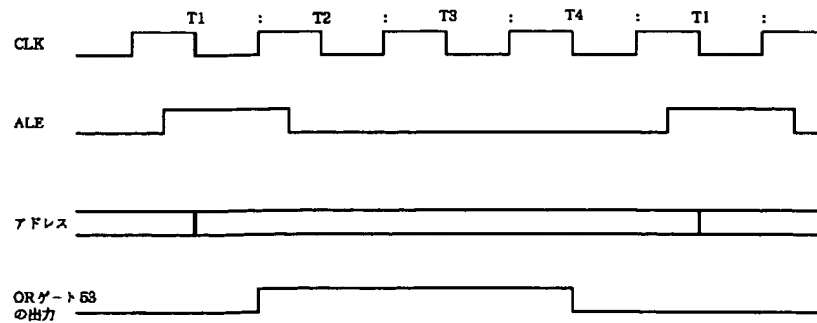
【図3】

104 OR AL,BL
106 OR AL,CL
10000 AND AL,BL
JMP 106

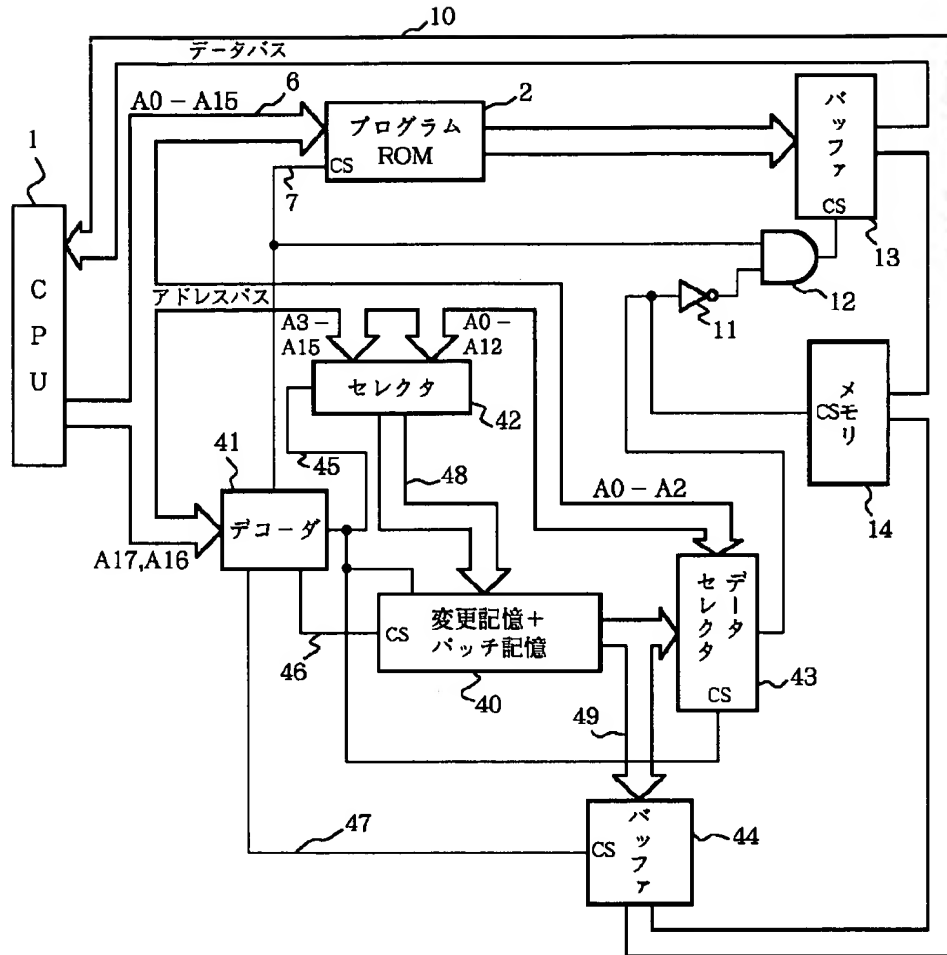
【図2】

| アドレス | 修正アドレス |
|------|--------|
| 104 | 10000 |
| | |
| | |

【図6】



【図4】



【図5】

